UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/691,295	10/22/2003	Eric Lawrence Barsness	ROC920030239US1	2239

30206 7590 05/02/2008
IBM CORPORATION
ROCHESTER IP LAW DEPT. 917
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER	
HICKS, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2165	
MAIL DATE	**DELIVERY MODE**
05/02/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/691,295
Filing Date: October 22, 2003
Appellant(s): BARSNESS ET AL.

Owen J. Gamon
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 3/3/2008 appealing from the Office action mailed 9/25/2007.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Sapia, "On Modeling and Predicting Query Behavior in OLAP Systems", Proceedings of the International Workshop on Design and Management of Data Warehouses, 1999

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-2, 6-9, 11-14, 16-17, and 20 rejected under 35 U.S.C. 102(b) as being anticipated by Sapia.

Note that Claim 16 recites all of the limitations found in Claims 1, 6, 9, 11, and 14, and Claim 2 recites all of the limitations found in Claims 8, 12, 13, and 17.

As per Claims 1, 3, 6, 9-11, 14, and 16, Sapia discloses a server, method, apparatus, and storage device comprising: a processor; and a storage device encoded

with instructions, wherein the instructions when executed on the processor comprise: finding a correlation between a first statement and a previous statement (i.e. *"If typical interaction patterns (task and/or user specific) are known, this information can be used to build query profiles for users or user groups. This information can be used to optimize the performance of an OLAP system at runtime. The idea is to use the profiles together with the know prefix of the query session to predict which queries the user is likely to ask during the rest of the session. Thus these queries or parts of the results can already be computed while the user is busy formulating his next query."* The preceding text excerpt clearly indicates that a correlation if found between queries of a present session (e.g. the first statement) and queries stored in a profile (e.g. a previous statement).) (Page 8, Column 1, Paragraph 2), wherein the previous statement is stored in a history of a plurality of statements (i.e. *"The profile information stores the user/task specific profiles (using the formalism presented in section 4). These patterns are initially constructed during the conceptual user modeling process as described earlier in this paper. Another interesting alternative is to build, validate and adapt these patterns by analyzing the query logs. This task is carried out by the profile builder which uses data mining/pattern recognition techniques to derive new profiles or adapt existing profiles."* The preceding text excerpt clearly indicates that the previous statement is stored in a history/profile.) (Page 8, Column 2, Paragraph 3), and wherein the finding the correlation further comprises finding a host variable in the previous statement in a history that matches a host variable in the first statement (i.e. See Pages 6-7, Section 4.2, specifically Definition 4.6 for examples of how the correlation (e.g. distance) between current user queries and queries in the profile are calculated using host variables (e.g. attributes). Note that attribute similarity is one of the measure determining operations to transform the queries which indicate the query distance.), wherein first data supplied for the host variable in the first statement matches previous data associated with the host variable in the previous statement (i.e. See Pages 6-7, Section 4.2, specifically Definition 4.6 for examples of how the correlation (e.g. distance) between current user queries and queries in the profile are calculated using

host variables (e.g. attributes). Note that attribute similarity is one of the measure determining operations to transform the queries which indicate the query distance.), wherein the host variable in the previous statement and the host variable in the first statement comprise a variable in a host language that is set to a plurality of values in succession and submitted to a database (i.e. See Pages 6-7, Section 4.2, specifically Definition 4.6 for examples of how the correlation (e.g. distance) between current user queries and queries in the profile are calculated using host variables (e.g. attributes). Note that the attributes are written in a host language (e.g. the language used to design the database).), predicting a second statement based on the previous statement (i.e. *"The core of the architecture is a prediction unit, which is located between the query processor and the frontend tool. It has the same interface to the user interface as the query processor (e.g. MDX) and predicts the possible next queries of the user from the status of the current session (the queries asked so far) and the profile information. Following a predefined strategy it passed predicted queries to the query processor for speculative execution. The results are stored in a speculative result cache."* The preceding text excerpt clearly indicates that a second statement (e.g. next possible query) is predicted based on the first statement (e.g. a query which was asked during the session).) (Page 8, Column 2, Paragraph 3; Page 9, Column 1, Paragraph 1), wherein the predicting further comprises finding the second statement that was next in time following the previous statement in the history (i.e. *"If not, the query is passed to the query scheduler which is responsible for passing the query on to the OLAP query processor. In any case the session state for the current session is updated (i.e. the current query prototype is added to the session prefix). This information together with the profile information is used by the 'prediction model'-process which generates queries that are most likely to be asked next and estimates their probability. In this process it makes use of the distance measure defined in section 4 as an approximation of similarity."* The preceding text excerpt clearly indicates that the predicted statement/query is next in time following the previous statement in the history/profile. Note the description of the distance measure (Definition 4.6).) (Page 9, Column 1, Paragraph 2), wherein the

previous statement and the second statement comprise commands that were previously executed against a database (i.e. *"The profile information stores the user/task specific profiles (using the formalism presented in section 4). These patterns are initially constructed during the conceptual user modeling process as described earlier in this paper. Another interesting alternative is to build, validate and adapt these patterns by analyzing the query logs. This task is carried out by the profile builder which uses data mining/pattern recognition techniques to derive new profiles or adapt existing profiles."* The preceding text excerpt clearly indicates that the previous statement and the second statement were both queries/commands that were previously executed in aganst the database, either by a user (e.g. query log method) or during the user modeling process (see Section 4).) (Page 8, Column 2, Paragraph 3), executing the first statement against the database (i.e. *"A cost model process passes the estimated database execution costs of the queries to a decision model process which decides if the query should be executed speculatively. In this case it passes the query to the query scheduler with a flag that the results of this query should be stored in the cache instead of being sent back to the user immediately. When the query scheduler receives a new query for execution, it first checks if such a query is already being executed at the moment. In this case the query is not passed on but answered using the results of the running query. This case can occur if a query that is being speculatively executed is actually asked by the user before the execution is finished. The results of all queries are either passed to the user or stored in the speculative result cache. The cache uses an appropriate replacement strategy (e.g. a time stamp method)."* The preceding text excerpt clearly indicates that the first statement (e.g. a statement which is not being speculatively executed) will be executed against the database and the results returned to the user.) (Page 9, Column 1, Paragraph 3, Column 2, Paragraph 1), retrieving at least one page from a database based on the second statement (i.e. *"A cost model process passes the estimated database execution costs of the queries to a decision model process which decides if the query should be executed speculatively. In this case it passes the query to the query scheduler with a flag that the results of this query should be stored in the cache instead of being sent back to the user immediately. When the query scheduler receives a new query for execution, it first checks if such a query*

is already being executed at the moment. In this case the query is not passed on but answered using the results of the running query. This case can occur if a query that is being speculatively executed is actually asked by the user before the execution is finished. The results of all queries are either passed to the user or stored in the speculative result cache. The cache uses an appropriate replacement strategy (e.g. a time stamp method)." The preceding text excerpt clearly indicates that the second statement (e.g. a statement which is being speculatively executed) will be executed against the database and the results (e.g. the at least one page) will be stored in a cache.) (Page 9, Column 1, Paragraph 3, Column 2, Paragraph 1), wherein the retrieving further comprises executing the second statement against the database (i.e. *"A cost model process passes the estimated database execution costs of the queries to a decision model process which decides if the query should be executed speculatively. In this case it passes the query to the query scheduler with a flag that the results of this query should be stored in the cache instead of being sent back to the user immediately. When the query scheduler receives a new query for execution, it first checks if such a query is already being executed at the moment. In this case the query is not passed on but answered using the results of the running query. This case can occur if a query that is being speculatively executed is actually asked by the user before the execution is finished. The results of all queries are either passed to the user or stored in the speculative result cache. The cache uses an appropriate replacement strategy (e.g. a time stamp method)."* The preceding text excerpt clearly indicates that the second statement (e.g. a statement which is being speculatively executed) will be executed against the database and the results (e.g. the at least one page) will be stored in a cache.) (Page 9, Column 1, Paragraph 3, Column 2, Paragraph 1), storing the at least one page in a cache (i.e. *"A cost model process passes the estimated database execution costs of the queries to a decision model process which decides if the query should be executed speculatively. In this case it passes the query to the query scheduler with a flag that the results of this query should be stored in the cache instead of being sent back to the user immediately. When the query scheduler receives a new query for execution, it first checks if such a query is already being executed at the moment. In this case the query is not passed on but answered using the results of the running query.*

This case can occur if a query that is being speculatively executed is actually asked by the user before the execution is finished. The results of all queries are either passed to the user or stored in the speculative result cache. The cache uses an appropriate replacement strategy (e.g. a time stamp method)." The preceding text excerpt clearly indicates that the second statement (e.g. a statement which is being speculatively executed) will be executed and the results (e.g. the at least one page) will be stored in a cache.) (Page 9, Column 1, Paragraph 3, Column 2, Paragraph 1), and executing a next statement against the at least one page in the cache (i.e. *"In figure 8 the dataflow and the processes inside the prediction unit are shown (the parts where the formalism presented in this paper is used are marked gray). The frontend passes a new query to the request handler. The handler checks if this query can be answered from the speculative result cache, i.e. has been correctly predicted."* The preceding text excerpt clearly indicates that a next statement (e.g. a query which is entered after the first statement, and after the result of the second statement has been placed in the cache) is executed and a check is made to see if the result of the next statement exists in the cache. As such, the next statement is executed against the page (e.g. result) in the cache.) (Page 9, Column 1, Paragraph 2), wherein the next statement follows the first statement in time (i.e. *"In figure 8 the dataflow and the processes inside the prediction unit are shown (the parts where the formalism presented in this paper is used are marked gray). The frontend passes a new query to the request handler. The handler checks if this query can be answered from the speculative result cache, i.e. has been correctly predicted."* The preceding text excerpt clearly indicates that a next statement (e.g. a query which is entered after the first statement, and after the result of the second statement has been placed in the cache) is executed and a check is made to see if the result of the next statement exists in the cache.) (Page 9, Column 1, Paragraph 2), and wherein the host variable in the next statement matches the host variable in the second statement (i.e. *"In figure 8 the dataflow and the processes inside the prediction unit are shown (the parts where the formalism presented in this paper is used are marked gray). The frontend passes a new query to the request handler. The handler checks if this query can be answered from the speculative*

result cache, i.e. has been correctly predicted." The preceding text excerpt clearly indicates that a next statement (e.g. a query which is entered after the first statement, and after the result of the second statement has been placed in the cache) is executed and a check is made to see if the result of the next statement exists in the cache. Note that in order to have the same result sets (e.g. in order for the result of the next statement to exist in the speculative cache) the next statement and the second statement must share host variables.) (Page 9, Column 1, Paragraph 2).

As per Claims 2, 8, 12, 13, and 17, Sapia discloses the retrieving further comprises: retrieving the at least one page asynchronously from executing the first statement against the database and storing the at least one page in a cache (i.e. *"The core of the architecture is a prediction unit, which is located between the query processor and the frontend tool. It has the same interface to the user interface as the query processor (e.g. MDX) and predicts the possible next queries of the user from the status of the current session (the queries asked so far) and the profile information. Following a predefined strategy it passed predicted queries to the query processor for speculative execution. The results are stored in a speculative result cache...A cost model process passes the estimated database execution costs of the queries to a decision model process which decides if the query should be executed speculatively. In this case it passes the query to the query scheduler with a flag that the results of this query should be stored in the cache instead of being sent back to the user immediately. When the query scheduler receives a new query for execution, it first checks if such a query is already being executed at the moment. In this case the query is not passed on but answered using the results of the running query. This case can occur if a query that is being speculatively executed is actually asked by the user before the execution is finished. The results of all queries are either passed to the user or stored in the speculative result cache. The cache uses an appropriate replacement strategy (e.g. a time stamp method)."* The preceding text excerpt clearly indicates that the one page (e.g. the result set) is retrieved asynchronously from the executing of the first

statement and that the one page (e.g. result set) is stores in a speculative result cache.) (Page 8, Column 2, Paragraph 3; Page 9, Column 1, Paragraphs 1, 3, Column 2, Paragraph 1).

As per Claim 7, Sapia discloses means for saving the first statement in the history (i.e. *"Another interesting alternative is to build, validate and adapt these patterns by analyzing the query logs. This task is carried out by the profile builder which uses data mining/pattern recognition techniques to derive new profiles or adapt existing profiles...Another interesting point is the derivation of user/task profiles from the saved interaction logs using data mining and pattern recognition techniques."* The preceding text excerpt clearly indicates that the queries which are used in the session may be saved to query logs and later inserted into the profiles/history.) (Page 8, Column 2, Paragraph 3; Page 9, Column 2, Paragraph 4; Page 10, Column 1, Paragraph 1).

As per Claim 20, Sapia discloses the finding the correlation further comprises:

finding the previous statement, wherein the previous statement is associated with a same job as the first statement (i.e. *"If typical interaction patterns (task and/or user specific) are known, this information can be used to build query profiles for users or user groups. This information can be used to optimize the performance of an OLAP system at runtime. The idea is to use the profiles together with the know prefix of the query session to predict which queries the user is likely to ask during the rest of the session. Thus these queries or parts of the results can already be computed while the user is busy formulating his next query."* The preceding text excerpt clearly indicates that the previous statement (e.g. the query found in the user profile) and the first statement (e.g. the known prefix f the query session) are associated with the same job (e.g. are used in the similar query sessions or are task specific).) (Page 8, Column 1, Paragraph 2).

(10) Response to Argument

As per Appellants arguments regarding the rejection of Claims 1-2, 6-9, 11-14, 16-17, and 20, Examiner respectfully disagrees.

In regards to Applicants assertions that the limitation of the host variables in the first and second statements matching wherein a host variable comprises a host variable in a host language that is set to a plurality of values in succession and submitted to the database, Examiner would like to further explain the sections of Sapia relied upon in the rejection to disclose this limitation in order to clarify Examiners position. In the cited section 4.2 of Sapia, Sapia discloses that the similarity between two queries is

determined, as pointed out by Appellant, by determining the number of operations needed to transform the query prototype of query q1 (e.g. the current user query) into the query prototype of query q2 (e.g. the previously submitted query). In order for it to be shown that this teaching further discloses the use of host variables as defined in the claims, it must first be established what is included in the query prototype. Sapia defines the query prototype in section 4.1, which, in the Example found in Definition 4.1, discloses that the query prototype includes references to the tables (e.g. vehicle.all) and attributes (e.g. year) which must be referenced to process the query and also indicates the purpose this information will serve in the query (e.g. will the information act as the query result, define selection granularity, etc.). Examiner notes that tables and attributes which are used to create the query prototype are drawn directly from the query itself, and are represented, when submitted to the database in the query, as data stored in a variable, which indicate to the database which tables and attributes to perform the indicated operations on. Examiner further notes that variable will be represented in the host language of the database (e.g. SQL) and that it is common to the art for database queries to support range type queries, which will be processed by the database as a series of queries in which the variable indicated as a range will be set to a plurality of values in succession and submitted to the database, after which the results of the series of queries representing the range will be combined and returned to the user as a result. As per the definition of the query prototype, it can be seen that in order to determine the distance between two queries the variable information contained in the query prototypes, which was drawn from the original queries, is compared, with

more matching variables performing similar functions indicating correlation between the two queries. As this is the case, Examiner asserts that limitations the inclusion and comparison of host variables, as claimed, is disclosed in Sapia.

As per Appellants arguments asserting that Sapia fails to disclose the limitation of finding the previous statement in the history and finding the second statement that was next in time following the previous statement in the history, Examiner notes that Appellants arguments fail to take into consideration how the distance measure referenced in Sapia is utilized to estimate the probability of queries which are likely to be asked next and asserts that the while the distance measure does determine the number of user interactions needed to transform one query into another in order to determine correlation, this does not preclude Sapia from teaching that a second query which is next in time from the previous query is identified as likely to be executed next. It is Examiners assertion that Sapia discloses an invention which first uses the distance measure to correlate the users current query (e.g. first statement) in the users current query session with a previously executed query identified from a query log (e.g. previous statement) which was executed in a previous user session, and then uses the previous sessions query sequence to predict the next query to be executed by identifying the query that immediately succeeds the correlated second query in the query sequence. Examiner notes several passages from Sapia to support this assertion. Firstly, on Page 2-5, Section 4.1, Sapia discloses "We are interested in modeling typical interaction patterns in query behavior. These patterns describe the similarities between different sessions that are similar with respect to the intention/task the user had in mind when executing the

session. From a system's point of view, the current interest (or intention) of the user is mirrored in the structure of the query he issues...", which indicates that the intent of the invention is to find similar user sessions by analyzing user queries from the current user session. This idea is supported by Page 6, Section 4.2 which indicates "We are interested in recognizing and formulating patterns of access. Two sessions have a common pattern if they contain similar queries. Thus, to define a pattern we first must define the notion of similarities of queries. We approximate the similarity of two queries q1 and q2 by defining a distance measure that corresponds to the number of user interactions minimally needed to transform the query prototype of query q1 to the query prototype of query q2. A short distance means a greater measure of similarity..." and Page 8, Section 5, which indicates, "The idea is to use the profiles together with the known prefix of the query session to predict which queries the user is likely to ask during the rest of the session." Examiner notes that the user profile is merely a collection of user specific query sessions, as indicated in Section 4.3 and that the sessions and profile may be built from query logs as noted On Page 2-5, Section 4, and Page 2-8, Section 5. Examiner asserts that the above passages clearly state Sapias intent to use current user queries to identify similar (e.g. correlated) queries in previous user sessions in order to predict future queries. Sapia further discloses the intent to then use the previous sessions query sequence to predict the next query to be executed by identifying the query that immediately succeeds the correlated second query in the query sequence on Pages 2-2, Final Paragraph and 2-3, First Paragraph, which state "To identify savings potentials a controller may start by accessing a predefined report (i.e. a query) ranking he geographical regions by number of repairs during the year 1998. He then picks a region in which enough repairs occurred to make it worthwhile further investigation. This is a cognitive process which takes some time and does not involve any system interaction. Therefore, from a systems point of view the process is a 'black box' that is not deterministic

but predictable under certain circumstances. The only thing the system can perceive (and use to guess the users intention) is the users next query..." (emphasis added by Examiner). Examiner notes that within the previously established context of using query logs to build sessions for query prediction and speculative execution, this clear statement of intention to use the users next query to guess (e.g. predict) the users intent clearly show that the query which is identified by the query prediction process and subsequently speculatively executed, as per Pages 2-8 and 2-9, is the next query which appears in the query log after the identified previous query. Page 2-3, Section 2 also supports this point by stating, in criticism of related work in the field, both that, "The approach does not that into account the correlation between successive queries due to the navigational nature of OLAP applications which is the basic assumption of our work..." and, "However, it does not model the distinction between selections and result dimensions, the relationship between consecutive queries, typical query patterns and does not contain a graphical presentation for query profiles." Examiner notes that Sapia clearly states that taking into account correlation between successive queries is a basic assumption of the method disclosed therein, and further notes that this directly supports Examiners above stated assertion.

In light of the above argument, Examiner asserts that the Sapia reference, when taken as a whole, teaches every limitation claimed by Appellant, including those argued in the instant Appeal Brief.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Examiner Michael Hicks
TC 2100
AU 2165

Conferees:

CHRISTIAN CHACE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Christian Chace

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER

Eddie Lee